Authors:	Järvelin Kalervo, Ingwersen Peter, Niemi Timo
Name of article:	A User-oriented Interface for Generalised Informetric Analysis Based on Applying Advanced Data Modelling Techniques
Year of publication:	2000
Name of journal:	Journal of Documentation
Volume:	56
Number of issue:	3
Pages:	250-278
ISSN:	0022-0418
Discipline:	Natural sciences / Computer and information sciences
Language:	en
School/Other Unit:	School of Information Sciences

URN: http://urn.fi/urn:nbn:uta-3-747
DOI: http://dx.doi.org/10.1108/EUM0000000007115

A USER-ORIENTED INTERFACE FOR GENERALIZED INFORMETRIC ANALYSIS BASED ON APPLYING ADVANCED DATA MODELLING TECHNIQUES

Kalervo Järvelin[+], Peter Ingwersen* and Timo Niemi[#]
[+]Dept. of Information Studies
[#]Dept. of Computer Science
University of Tampere
P.O.Box 607
FIN-33101 TAMPERE, Finland

*Royal School of Library and Information Science
Birketinget 6
DK-2300 COPENHAGEN S, Denmark

ABSTRACT

This article presents a novel user-oriented interface for generalized informetric analysis and demonstrates how informetric calculations can easily and declaratively be specified through advanced data modeling techniques. The interface is declarative and at a high level. Therefore it is easy to use, flexible, and extensible. It enables end-users to perform basic informetric ad hoc calculations easily and often with much less effort than in the contemporary online retrieval systems. It also provides several fruitful generalizations of typical informetric measurements like impact factors. These are based on substituting traditional foci of analysis, for instance journals, by other object types, such as authors, organizations, or countries. In the interface, bibliographic data are modeled as complex objects (non-first normal form relations) and terminological and citation networks involving transitive relationships are modeled as binary relations for deductive processing. The interface is flexible, because it makes it trivial to switch focus between

various object types for informetric calculations, e.g. from authors to institutions. Moreover, it is demonstrated that all informetric data can easily be broken down by criteria that foster advanced analysis, e.g., by years or content-bearing attributes. Such modeling allows flexible data aggregation along many dimensions. These salient features emerge from the query interface's general data restructuring and aggregation capabilities combined with transitive processing capabilities. The features are illustrated by means of sample queries and results in the article.

1. INTRODUCTION

Informetrics studies various statistical phenomena of literature often based on bibliographic information provided by online databases. Among the statistical phenomena are productivity issues of authors, countries, or journals [1, 2] and generalized impact factors of journals or authors [3, 4]. Also activity profiles of authors, organizations, and journals, or citation networks in the form of bibliographic coupling of authors or articles and author co-citation analysis [5] as well as literature growth and aging can be computed [6].

Several informetric measurements are produced by the ISI (Institute of Scientific Information), published in their reports, e.g., the Journal Citation Report. Informetric calculations can also be done online in the online databases. Hjortgaard Christensen, Ingwersen and Wormell [4, 7] have described the methodology of various citation-based analyses using the One-Search, RANK and TARGET commands of the Dialog Information Service. Very often ad hoc informetric measurements are needed for decision making, e.g., for competitor information, science policy, research project funding, etc.

This article considers how informetric measurements can easily and declaratively be specified through data management techniques. We consider typical informetric data, i.e., bibliographic data as well as terminological and citation networks involving transitive relationships. The poor capability of the conventional relational model in modeling and processing complex objects in many applications, including information retrieval (IR), has led many researchers to study the NF^2 relational model, i.e. non-first normal form relations [8], object-oriented databases, deductive databases and deductive object-oriented databases (e.g., [9, 10, 11]). Bibliographic data are naturally modeled as NF^2 relations [12, 13]. Moreover, some terminological network structures, e.g., thesauri and classifications, and citation networks are not complex objects but rather represent transitive relationships and cannot therefore be modeled by the NF^2 principle. Thus we shall model such structures as binary relations which support computations involving transitive relationships [14, 15, 16]. The paper demonstrates how the management of NF^2 relations and transitive relationships is integrated.

We shall introduce briefly a very high-level declarative query interface based on NF^2 relations and transitive relationships [13, 15]. This interface, called the FUN interface, provides general data restructuring and aggregation capabilities combined with general transitive processing capabilities thus providing powerful features for retrieving and analyzing bibliographical and citation data. The need for such capabilities has been recognized as necessary in many document and IR related studies (see, e.g., [17, 18, 19]). The data aggregation capability of online IR systems falls short for several informetric measurements. For example, Dialog's Rank feature [20] and ESA-IRS's Zoom feature [21] merely provide term counts in a single field of a retrieved set of documents. Persson's recently developed bibliometric toolbox, available on the Net, is limited to productivity data only [2]. We shall show that general data restructuring and multi-level aggregation are necessary for informetrics. In addition to these general capabilities, an essential feature of the FUN interface is its very high abstraction level

and declarativity. The user need not specify how the results are derived from the database. Instead, the interface deduces the derivation steps even in complex query situations.

This article demonstrates that the proposed data modeling and query interface enable end-users to perform basic informetric ad hoc calculations, such as generalized impact factors, author co-citation analysis, productivity calculations in a given area, etc., easily and often with much less effort than in contemporary online retrieval systems. For instance, users need not determine in advance a set of all author pairs for co-citation analysis and derive the data separately for each pair — this is done by a single query. We shall also propose several fruitful generalizations of typical informetric measurements. They are based on substituting traditional foci of analysis, for instance journals, by other object types, such as authors, organizations, countries or classes of a classification scheme. It is shown that the FUN interface makes it simple to switch focus between various object types for informetric calculations. Moreover, it is demonstrated that all informetric data can easily be broken down along several dimensions that foster advanced analysis, e.g., by years or by content-bearing attributes. Thus, our data modeling and query interface support generalized informetrics. As a spin-off effect, citation data may be used for IR purposes. This is an area of IR research that has been neglected in recent years.

Ingwersen and Hjortgaard Christensen [22] have pointed out that the consistency of database contents is essential for informetric analysis. In this paper we shall not consider the problems caused by real bibliographic databases containing corrupted, incomplete data, and partially incompatible data, e.g., varying journal names in citations. Instead, we shall utilize a small bibliographic sample database, not suffering from such problems, for our analyses. In practice, our interface is dependent on the quality of downloaded data. However, this is a problem to be considered also in all other approaches. The quality problems are no worse for our approach than in the traditional online or offline situations.

2. SAMPLE DATABASE ENVIRONMENT

Figures 2.1 - 2.4 exemplify three data modeling situations where two kinds of modeling principles are necessary. Figures 2.1 a-b show the data structure diagram and a sample instance of a complex object, or NF^2 relation, representing bibliographic references. In the diagram, rectangles represent relation-valued attributes while ellipses represent the atomic-valued attributes (or properties) of each relation-valued attribute. Thus the complex object ARTICLES has two levels with the relation-valued attribute ARTICLES forming the *top relation,* and the relation-valued attributes AUTHORS, KEYWORDS and CLASSES forming its *immediate subrelations.* The latter relation-valued attributes are *bottom relations.* The sample instance in Figure 2.1. (b) displays five articles from three different journals, having one or more authors (with affiliations), and several keywords as well as several classes. Complex objects of type ARTICLES are structurally static in the sense that all objects have exactly two levels. No recursive structure is present. Complex objects of type ARTICLES are formed from more simple objects of various types and are naturally represented by NF^2 relations.

The relation ARTICLES has the atomic-valued attributes *ano* (article number), *title* (article title), *publisher* (publisher number), *journal* (journal name), *year* (publication year), *vol* (journal volume), *issue* (journal issue) and the three relation-valued attributes AUTHORS, KEYWORDS and CLASSES. These latter three attributes contain the atomic-valued attributes *author* (article author), *department, organization, city and country* (which give the author's affiliation), *key* (a thesaurus term) and *class* (a class of the ACM Computer Science Classification), respectively. NF^2 relations are excellent for modeling structurally static complex objects such as the relation ARTICLES. They are not suited for modeling structurally dynamic objects like thesauri or citation networks which have, in principle, unlimited transitive relationships between nodes.

In some contemporary public databases, for instance, the citation databases produced by the Institute for Scientific Information (ISI), there is no direct link between each author and his/her affiliation. Such a limitation in the input data must be resolved prior to data analysis for all queries that require the affiliation data per author.

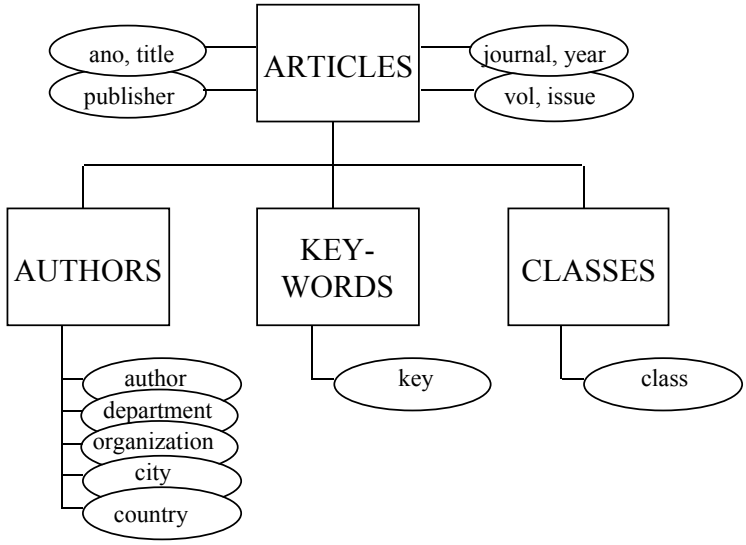


Fig. 2.1 (a) Modeling bibliographic references as complex objects: the data structure diagram

```
{(art_1, The relational model in information retr, John Wiley & Sons,
  JASIS, Journal of the American Society f..., 32, 1, 1980,
  {(Crawford, R, Department of Computing ..., Queens Univer..., Kingston, Canada)},
  {(relational database), (sequel), (relational algebra),(bibliographic database)},
  {(H.2.1), (H.3.3)}),

 (art_3, Universal relation theory applied to bib, The Canadian Association for
  Information, The Canadian Journal of Information Scie, 9, 1, 1984,
  {(Crawford, R, Department of Computing ..., Queens University, Kingston, Canada),
   (Becker, S, Department of Computing ..., Queens University, Kingston, Canada),
   (Ogilvie, J, Department of Computing ..., Queens Univer..., Kingston, Canada)},
  {(relational database), (universal relation), (bibliographic database)},
  {(H.2.1), (H.3.3)}),

 (art_5, Non-first normal form universal relation, Pergamon, Information Systems,
  12, 1, 1987,
  {(Desai, B, Department of Computer Sci…, Concordia University, Montreal, Canada),
   (Sadri, F, Department of Computer Sci…, Concordia University, Montreal, Canada),
    (Goyal, P, Department of Computer Sci…, Concordia University, Montreal, Can-
ada)},
  {(non-first normal form relation), (universal relation),…, (document retrieval)},
  {(H.2.3), (H.2.4), (H.3.3)}),

 (art_20, Deductive Information Retrieval Based on, John Wiley & Sons,
  JASIS, Journal of the American Society f..., 44, 10, 1993,
  {(Niemi, T, Department of Computer Sc…, University of Tampere, Tampere, Finland),
   (Jarvelin, K, Department of Inf… St…, University of Tampere, Tampere, Finland)},
  {(query languages), (knowledge-based retrieval), (deductive database), ...},
  {(H.2), (H.3.2), (H.3.3), (I.2.4)}),

 (art_21, Text Retrieval and the Relational Model, John Wiley & Sons,
  JASIS, Journal of the American Society f..., 42, 3, 1991,
  {(Macleod, I, Department of Computing …, Queens University, Kingston, Canada)},
  {(relational database), (text retrieval), (query languages)},
  {(H.2.2), (H.2.1), (H.3.2), H.3.3)}),
... }
```

Fig. 2.1 (b) Modeling bibliographic references as complex objects: partial instance

Our query interface employs a linear data structure representation called *form.* A form gives the relation-valued and atomic-valued attribute names of a relation and employs parentheses to denote the nesting level of each component. The form ARTICLES(ano, title, publisher, journal, year, vol, issue, AUTHORS(author, department, organization, city, country), KEYWORDS(key), CLASSES(class)) corresponds to the data structure diagram of Figure 2.1a. We follow the convention of marking relation-valued attribute names in capital letters and atomic-valued attribute names in lower case letters.

Figure 2.2 shows the data structure diagram and a sample instance of the thesaurus TERM objects and their thesaural relationships. Each TERM object is atomic, i.e., there is only the Term-Name attribute in each object. The data structure diagram shows that thesaurus terms are related to themselves through the subterm (ST, SUBTERM) relationship. The relationship

SUBTERM is transitive, i.e., if document retrieval is an *immediate subterm* of data management and query formulation is an *immediate subterm* of document retrieval, then query formulation is a *transitive subterm* of data management. The sample instance shows an excerpt of terms in the hierarchic SUBTERM relationship.

Thesaurus objects are structurally dynamic in the sense that they have unlimited acyclic transitive relationships with varying depth in different directions from any given TERM object, i.e., the structure is recursive. Thesaurus-like structures cannot be modeled as structurally static complex objects, like NF^2 relations. They can, however, be modeled through binary relations representing transitive relationships indirectly.

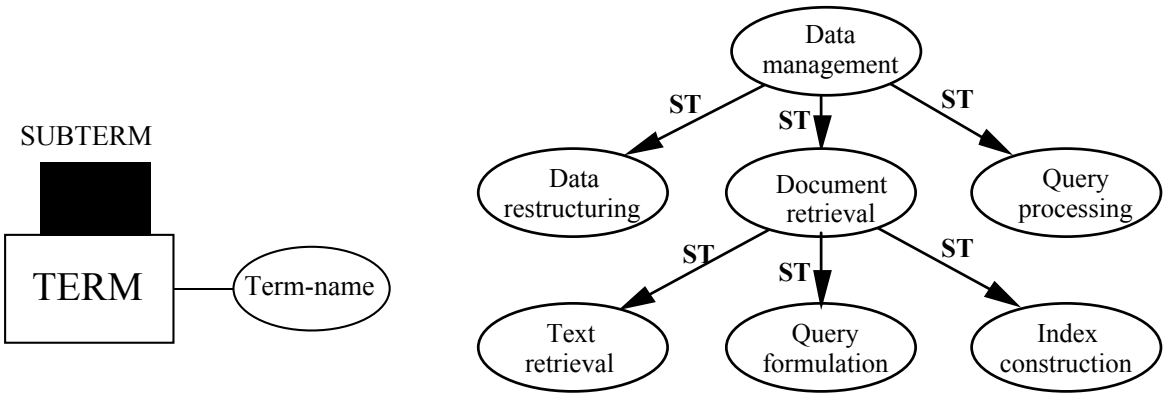


Fig. 2.2. Modeling a transitive hierarchic relationship

Figure 2.3 shows the data structure diagram and a sample instance of simple ARTICLE objects and their citation relationships. Each ARTICLE object is atomic, i.e., there is only the *ano* attribute in each object. The data structure diagram shows that articles are related to themselves through the transitive citation (CITES) relationship. The sample instance shows an excerpt of a citation network. Also citation networks are structurally dynamic in the sense that they have unlimited acyclic transitive relationships with varying depth from any given ARTICLE object.

The reference list of an article — the cited articles — could well be represented as a relation-valued attribute of an article in an NF^2 relation. However, this would not support finding *citing articles* of a given article. Thus citation networks are not usefully modeled as structurally static complex objects, like NF^2 relations.

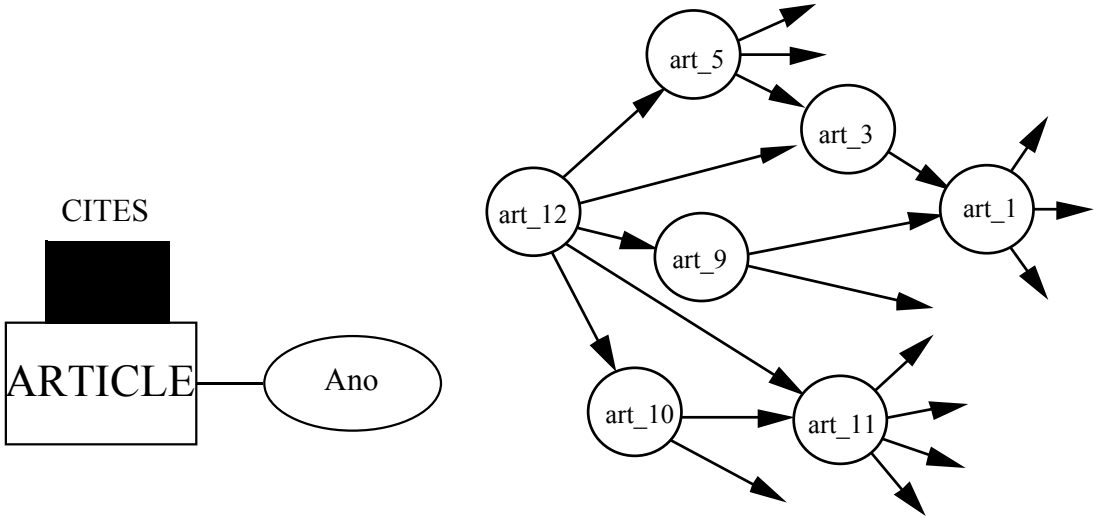


Fig. 2.3. Modeling a transitive non-hierarchic relationship

SUBTERM	
PREDECESSOR	**SUCCESSOR**
Data management	Data restructuring
Data management	Document retrieval
Data management	Query processing
Document retrieval	Query formulation
Document retrieval	Text retrieval
Document retrieval	Index construction

CITES	
PREDECESSOR	**SUCCESSOR**
art_12	art_5
art_12	art_3
art_12	art_9
art_12	art_11
art_12	art_10
art_5	art_3

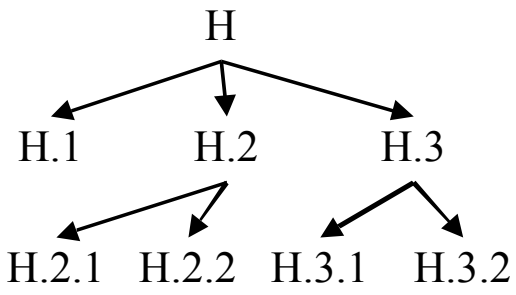
Fig. 2.4. Representing transitive relationships as binary relations (partial)

Figure 2.4 shows how the transitive relationships are represented as binary relations. The columns are labeled as PREDECESSOR and SUCCESSOR. In the case of the hierarchic term relationship SUBTERM, predecessors give the hierarchically higher terms and successors the

hierarchically lower terms. In the case of the citation relationship CITES, predecessors give the *citing articles* and successors the *cited articles* (NB: predecessor is later in time). The transitively hierarchically lower terms of, e.g., 'Data management' are denoted by successors('Data management', [SUBTERM]) = {Data restructuring, Document retrieval, Query processing, Query formulation, Text retrieval, Index construction}. The first argument specifies the starting object and the second the binary relation as the context of transitive computation. Similarly, the immediate citing articles of, for instance, article art_1 are denoted by im_predecessors(art_1, [CITES]) = {art_3, art_9, ... }, Figure 2.3. The immediate *cited articles* (i.e., references) of article art_5 are denoted by im_successors(art_5, [CITES]) = {art_3, ...}. These notations correspond to the operations of our query language for transitive processing [16, 23] which will be used below. The query language also has an acyclicity checking tool for the binary relations.

The citation relationship CITES contains also self-citations, i.e., one of the authors of the citing document belongs to the authors of the cited document. In some analyses this would distort the statistics and therefore we sometimes use a subset CITES2 of the citation relationship CITES from which self-citations have been excluded.

Figure 2.5 shows a transitive hierarchic relationship, in this case the Computer Science Classification. All subclasses of, for instance, the class H.3 are denoted by successors(H.3, [SUBCLASS]) = {H.3.1, H.3.2}.



SUBCLASS	
PREDECESSOR	**SUCCESSOR**
H	H.1
H	H.2
H	H.3
H.2	H.2.1
H.3	H.3.1
H.3	H.3.2

Fig. 2.5. Representing transitive relationships of the CS Classification (partial)

In summary, an NF^2 relation-like complex object representation is not suitable for representing structures based on transitive relationships. Terminological relationships and citation (or other link-based) networks are not aggregation hierarchies in the data modeling sense [24]. The networks consist of instances of objects (nodes) of a single type. In processing transitive relationships, the management of indirect node relationships is of prime importance, not the structure of nodes. Complex objects are needed when several separate objects with their own identity are put together to represent a complex real world entity, such as a document. In processing complex objects, the management of structural relationships is of prime importance. There is no static structure among subdocuments (component objects) in which all users would always want their result documents, e.g., articles by journals or by institutions. Therefore a mechanism for restructuring the hierarchical relationships among subdocuments into new result documents is needed [25], e.g., articles by domains.

In informetrics, complex objects (like bibliographic references), hierarchic transitive relationships (thesauri), as well as non-hierarchic transitive relationships (citation relationships) are all needed and often in combinations. In the sample database there are data for six object types that are typical foci of informetric analysis: authors, articles, journals, departments and their parent organizations as well as countries. We shall demonstrate that it is very easy for

the user to obtain various informetric analyses of all these object types and, further, very easy to swap between the object types in the analyses. The database also contains several attributes typically used to select and/or break down the statistical data for trend analysis: keywords, classification codes, and publication years. Nothing prevents using object types (e.g., journals) for data breakdowns and the breakdown attributes (e.g., years or classes) as the objects to analyze. The data represented by complex objects and transitive relationships are integrated through queries explained in the next section.

3. THE QUERY INTERFACE

So far, the query languages proposed for novel database paradigms have been too cumbersome to use from the viewpoint of end-users: users are required to derive the result data from the existing data by, often recursive, logical rules or constructors. Large nested expressions are usual in queries that combine data aggregation, transitive computation and data restructuring [25].

The idea behind the FUN interface is that all required data manipulation operations are deduced automatically on the basis a high-level declarative query specification. The user only has to express seven simple constructs in query formulation, when full aggregation, restructuring, transitive processing, sorting and retrieval capabilities are needed. The FUN interface has been described in earlier publications [13, 15, 25]. A query in the FUN interface is structured according to the following constructs:

- the **form** construct is the linear schema representation of the result NF^2 relation,
- the **relations** construct is a list of names of existing (source) first normal form (1NF; [26]) or NF^2 relation(s) providing the source data for the query,

- the **conditions** construct is a Boolean expression which gives the filtering conditions of atomic-valued and relation-valued attributes,
- the **aggregation** construct is a list giving the aggregation way (e.g., sum, max) of each aggregated attribute,
- the **subquery** construct describes any transitive and other processing needed in the construction of each relation-valued result attribute
- the **sorting** construct is a list of atomic-valued attribute names used for sorting the result relation-valued attributes,
- the **printing** construct is a list of names of relation-valued attributes in the output.

The user gives these seven components in a straightforward way as exemplified below. Nothing else is required from the user. The query processing system deduces the retrieval, restructuring, aggregation and deductive operations needed for producing the result NF^2 relation from the source NF^2 relation(s). It also executes the expressions given in the **subquery**-component and applies the results according to the **condition** and/or the **form** constructs in the construction of the result. In the interface, the user specifies the schema level of the result NF^2 relation declaratively and the query processing system constructs its instance.

The FUN interface is structured in a conventional style, resembling SQL. However, there are several differences with respect to the proposed SQL extensions (see, e.g., [27, 28, 29]) for processing NF^2 relations. (i) Our interface does not contain any explicit restructuring expressions — all restructuring is specified implicitly in the form. (ii) Multi-attribute multi-way multi-level aggregation may be specified declaratively in a single query without nested expressions. (iii) Finally, transitive processing is integrated conveniently through available high-level operations in the **subquery**-component. Therefore, queries in the FUN interface remain compact also when complex processing is required.

The query processing strategy and implementation issues are described by Niemi and Järvelin [15, 30]. The FUN interface has been implemented in LPA Prolog and runs on PCs and Macintoshes, as well as in Quintus Prolog for Unix machines. The sample query results in the following section are output from the system using a small sample database.

In this paper we shall present a user interface for informetric computation, which is based on online dialogs. Using this interface, the user need not use directly the query language introduced above. This is important, because the high-level query language may still be too demanding for non-technical users and it is very easy to model repeating queries into simple online dialogs that fill in the variables for a query. Figure 3.1 presents the main menu dialog from which the user may choose various kinds of informetric analyses. Sample informetric queries in Section 4 will be presented mainly as online dialogs. The options in the pull down menu are 'Impact Factors', 'Cocitation Analysis', 'Recognized Contribution', 'International Visibility', 'International Impact', 'Productivity Analysis', and 'Bibliographic Coupling'.

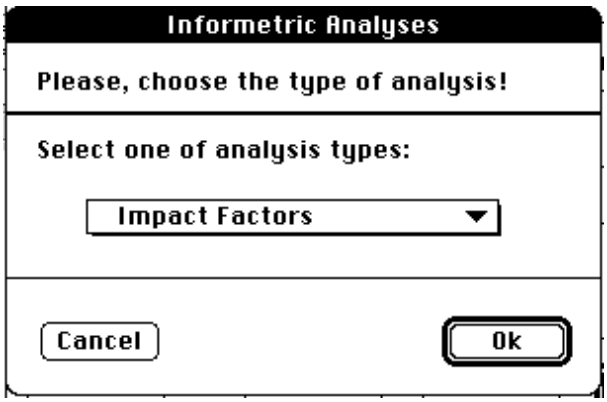


Fig. 3.1. The main menu dialog

4. INFORMETRIC QUERIES

4.1. Generalized impact factors

Journal impact factors are among the most important and popular citation analytic measures [3, 31]. They are used, e.g., in the assessment of the expected scientific merit of scholars or research groups. The Journal Citation Report by ISI is a standard source for journal impact factors. Hjortgaard Christensen and Ingwersen [7] demonstrate how various citation analyses of journals may be performed online, by using the Dialog retrieval system, for one or more volumes of a specific journal. The following remarks can be made concerning the state-of-the-art methodology presented recently by Wormell [32]:

- The user needs to process each journal separately.
- The user needs to specify each range of years of citation and publication separately.
- The resulting data require statistical post processing before the number of citations to each volume of each particular journal can be derived.

In this section we demonstrate how journal citation analyses, in particular journal impact factors, can be performed conveniently through the FUN interface. We shall also demonstrate how journal citation analyses are easily generalized to citation analyses of other object types, for instance, authors, institutions, countries, or classes of a classification. This is important since only authors, journals, and cited publication years at present can be analyzed directly for citation impact in the ISI citation databases. Figure 4.1 shows the dialog for impact factor analysis. By selecting the proper radio button, impact factors for journals, authors and institutes may be computed. Impact factors for classes have a separate dialog. The sample impact factor analysis below has the following verbal definition: the number of citations given during the period 1988-1995 to the articles of each journal in the database published in the pe

riod 1980-1990, divided by the number of citable articles published by each journal during the period 1980-1990.

Sample Expression 1 (see Figures 4.2 a-b) has two queries to avoid nested expressions in the **subquery** construct. The first query limits the *citation window* (years of publication of citing articles) to the desired years. In our sample case we use a relatively broad window (1988-1995), because the sample database is small. However, any window length can be used. This may often be a relevant way to generalize impact factors [4]. Also, any further conditions may be applied, e.g., the citing articles may be limited by journals, countries and/or disciplines. The **form** construct determines that the result is a flat relation 'CITWINDOW' consisting only of article numbers published within the time range [1988, 1995]. Because only the three first components of the expression for 'CITWINDOW' specify any processing, the remaining components have been omitted.

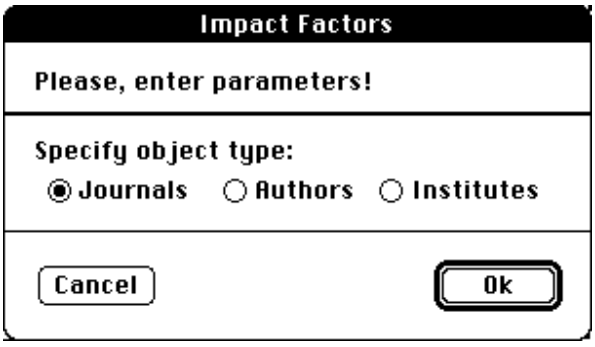


Fig. 4.1. The main menu for impact factor analysis

Sample Expression 1

```
CITWINDOW =
        form          CITWINDOW(ano)
        relations     ARTICLES
        conditions    year = between([1988, 1995])


form          JOURNAL(journal,
                  PUBLYEAR(year, citation_sum, art_cnt,
                        ARTS(ano, citation_cnt,
                              CITATION(citing_art))))
relations     ARTICLES
conditions    year ≥ 1980 and year ≤ 1990
aggregation   citation_sum = sum(citation_cnt);
              citation_cnt = cnt(citing_art)
              art_cnt = cnt(ano);
subquery      CITATION(citing_art) =
                  set_intersection(
                        im_predecessors(ano, [CITES]),
                        CITWINDOW)
sorting       journal, year
printing      JOURNAL, PUBLYEAR
```

Fig. 4.2 (a) Sample Expression 1 for journal impact factor calculation

The **form** construct of the main query specifies a data structure consisting of four levels of hierarchy. The top relation 'JOURNAL' gives journal names. For each journal, the relation-valued attribute 'PUBLYEAR' gives each year when the journal has published articles that are cited, together with citation statistics: the *sum of received citations* and the *number of citable articles*. Within each year, the relation-valued attribute 'ARTS' gives the citable articles of that year and the number of citations for each article. For each article number, the relation-valued attribute 'CITATION' identifies the citing articles. This relation-valued attribute is constructed by the **subquery** construct (see below). In this form the atomic-valued attributes 'journal', 'year' and 'ano' are *source relation attributes* and the rest *derived attributes.* Among the latter, 'citation_sum', 'art_cnt' and 'citation_cnt' are *aggregated attributes* and 'citing_art' a *deductive attribute* derived through a subquery.

The **conditions** construct of the main query specifies the *publication window* of the cited articles as the years within the range [1980, 1990]. The user may express any other conditions concerning the source relation attributes and/or derived attributes, for instance, conditions on cited article topics, cited journal names, etc. The **aggregation** construct states that the values of the aggregated attribute ‘citation_cnt’ are counts on the values of the attribute ‘citing_art’. Similarly, values of ‘citation_sum’ are sums of the values of the attribute ‘citation_cnt’ and values of ‘art_cnt’ are counts on the values of the attribute ‘ano’. Thus multiple attributes are aggregated at two levels at once.

The **subquery** in Figure 4.2(a) constructs the relation-valued attribute ‘CITATION’. The left-hand side of the expression is the form of the relation-valued attribute and the right-hand side expresses its derivation. The expression im_predecessors(ano, [CITES]) finds *all articles citing the individual articles* (each identified by the 'ano' -value) for which the relation-valued attribute 'CITATION' is being constructed. The result is a set of citing article numbers. The NF^2 relation name 'CITWINDOW' denotes the whole 'CITWINDOW' relation, the (citing) article numbers of which are returned as a set [23]. The two sets of article numbers are finally intersected by the operation set_intersection. This yields a set of numbers for articles that cite the article under consideration and are published within the citation window 1988-95. One should note that in formal scientific communication an *article* is only *cited once* on a reference list. However, the *journals* in question can be cited *several times* by the same article.

The **printing** construct of the main query specifies that only the two top relation-valued attributes 'JOURNAL' and 'PUBLYEAR' are reported as the result. The other relation-valued attributes are, in fact, only needed for computing the aggregated attributes and can therefore be omitted from the result. The **sorting** construct specifies that the relation-valued attribute 'JOURNAL' is sorted on journal names and the attribute 'PUBLYEAR' on years.

Figure 4.2b presents the standard dialog for journal impact factors. It allows impact factor calculations for any set of named journals as chosen by the user, or all journals in the database. The citation and publication window years can also be given. It is easy to modify these dialogs to accommodate other frequent parameters, for instance, countries or scientific domains, when needed. The journal name menu is constructed by a query in the FUN query language.

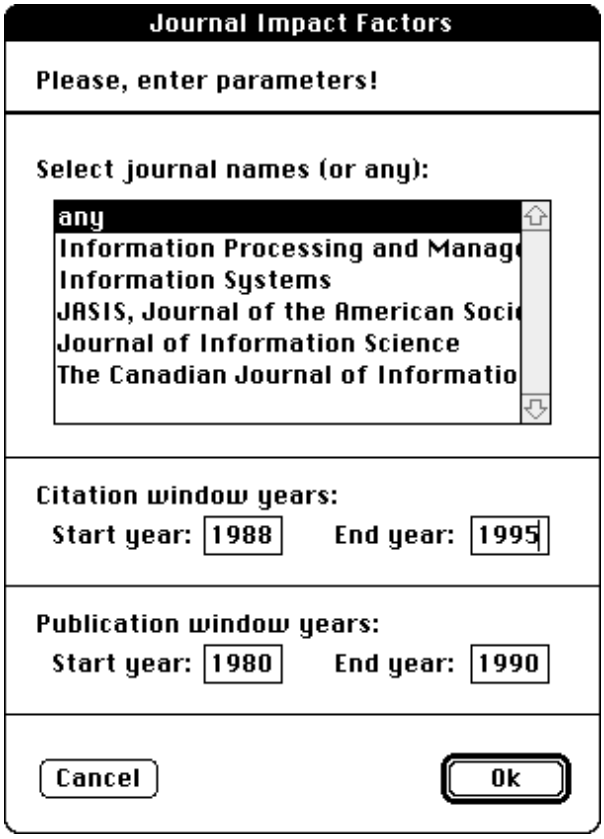


Fig. 4.2 (b) The online dialog for standard journal impact factors

The query result (Figure 4.2c) gives the 7-year synchronic impact data for five journals and various individual years within the range of 1980-90. For example, during the period 1988-95 JASIS has received four citations for one article published in 1980 (note that the sample database is small).

```
{(Information Processing and Manageme,
        {(1990, 1, 1)}),
 (Information Systems,
        {(1981, 1, 1),
         (1986, 1, 1),
         (1987, 2, 1)}),
 (JASIS, Journal of the American Soci,
        {(1980, 4, 1)}),
 (Journal of Information Science,
        {(1987, 1, 1)}),
 (The Canadian Journal of Information,
        {(1984, 1, 1)})}
```

Fig. 4.2 (c) Sample Expression 1 result: Impact factor data for journals

Sample Expression 1 has several salient features:

- The user need not process each journal separately. Instead, he gets data for all relevant journals automatically. Note that the **condition** construct could contain any conditions directly on journal names, publishers, countries of publication, and/or scientific domains combined with either the cited or the citing articles or both.
- The user need not specify each year of citation separately. Instead, he gets data for all relevant years automatically.
- The resulting data give the sum of citations as well as the number of citable articles directly for impact factor calculation. If required summations over the publication years for each journal are easily obtained by defining two new attributes, for example, sum_of_citations and sum_of_articles. See for instance the data for Information Systems.
- Multi-level multi-attribute aggregation is performed in a single query.

When the properties of citing and/or cited documents are used in the query, these documents must be included in the database as fully represented documents. In practice, all databases contain documents, which either give or receive citations across the database boundaries and thus the citing or cited documents are external to the database. However, this limitation affects all approaches to citation analysis.

4.1.1 Author impact factors

Through the FUN interface, it is very simple to obtain data for various generalizations of impact factors. Järvelin, Ingwersen and Niemi [33] discuss in detail at the query expression level the modification of expressions for the generalized impact factors and other informetric analyses. For example, for *author impact factors,* it is sufficient just to change the **form**, **sorting** and **printing** constructs as follows (changes in *italics*):

form *AUTHOR(author,* citation_sum, art_cnt,
ARTS(ano, citation_cnt,
CITATION(citing_art))))
sorting *citation_sum*
printing *AUTHOR*

The system developer need not do anything else and therefore it is very easy for him to provide users dialogs for analyzing the data in various ways. In this case we left out the data breakdown by years simply by dropping the relation-valued attribute 'IMPACTYEAR' and the attribute 'year', and by moving the aggregated attributes by one level up. From now on, we do not present formal query expressions but rather focus on the online dialogs for selected generalized informetric analyses and their sample results.

By selecting 'Authors' in the dialog of Figure 4.1, the dialog of Figure 4.3(a) for standard impact factors query for authors is presented. The user may specify the author set and the citation and publication window years. Here, all authors are selected for the publication window 1980-89 with the citation window 1990-95. From the same source data as above, the result is as given in Figure 4.3(b). For example, Crawford has received five citations during the period 1990-95 for the two articles he has in the database (published 1980-89).

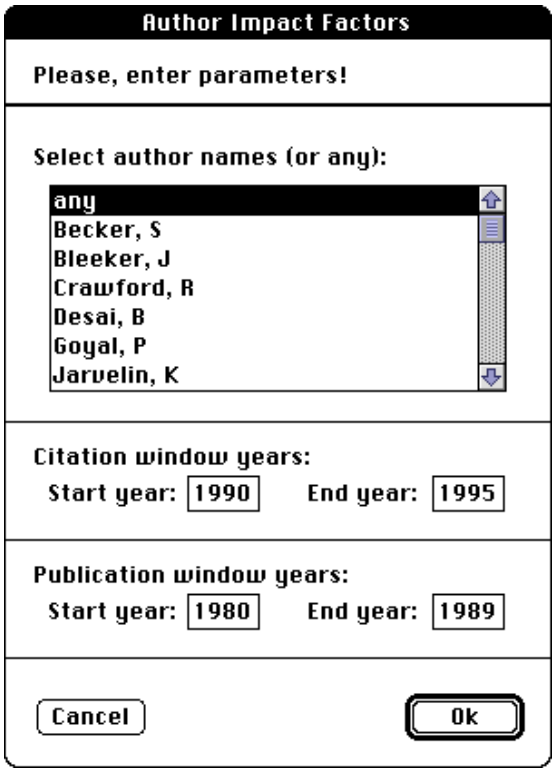


Fig. 4.3 (a) The online dialog for standard author impact factors

```
{(Becker, S,    1, 1),
 (Bleeker, J,   1, 1),
 (Crawford, R,  5, 2),
 (Desai, B,     2, 1),
 (Goyal, P,     2, 1),
 (Kircz, J,     1, 1),
 (Macleod, I,   1, 1),
 (Ogilvie, J,   1, 1),
 (Sadri, F,     2, 1),
 (Scheck, H,    1, 1),
 (Scholl, M,    1, 1)}
```

Fig. 4.3 (b) Impact factor data for authors

4.1.2 Institutional impact

The online dialog for standard institutional impact factors is similar in structure. Figure 4.4 presents the data for three selected institutes for citations given in 1990-97 to their publications in 1980-93. For example, Queens University has received eight citations during the period 1990-97 for the five articles published by the university in the database in 1980-93.

```
{(Concordia University,  2, 1),
 (University of Tampere, 4, 3),
 (Queens University,     8, 5)}
```

Fig. 4.4. Impact factor data for organizations

One may notice that, in contrast to the proposed NF^2 relational model, institutional or national impact factors *cannot* be obtained in contemporary CD-ROM or online citation indexes directly. They are only available through cumbersome selection of individual cited documents authored by the institution or country.

In a similar way, the impact factors can be computed for disciplines (if journals have discipline codes), topical classes, keywords, etc. Therefore we may conclude that it is very easy to build dialogs for various impact factors by simply manipulating the **form** construct. The required data breakdowns are obtained by placing source attributes in suitable positions within relation-valued attributes of the form. The data for the immediacy index, another popular informetric measure, and its generalizations may be obtained in a very similar way.

4.2. Productivity calculations

The productivity data of journals in a given topical area form the basic data for (i) impact factor calculations in the form of the denominator; (ii) Bradford's law of scattering (e.g., [34]). The journal productivity figures may be computed by the online dialog for standard journal productivity, Figure 4.5(a). Through the two pop-up menus the user may select among object types author (cf. Lotka's law on publication productivity per scientist), journal, institution and country, and among ACM CS Classes as domains of productivity. However, any available classification or thesaurus may easily be integrated — even several alternative ones, if desired. The publication window may be selected through the edit fields as a range of years. In this case we consider *journal productivity* for articles belonging to the study of "in

formation retrieval" (ACM CS Class 'H.3') published in 1990-97. Ideally, the productivity result should display a Bradford-like distribution.

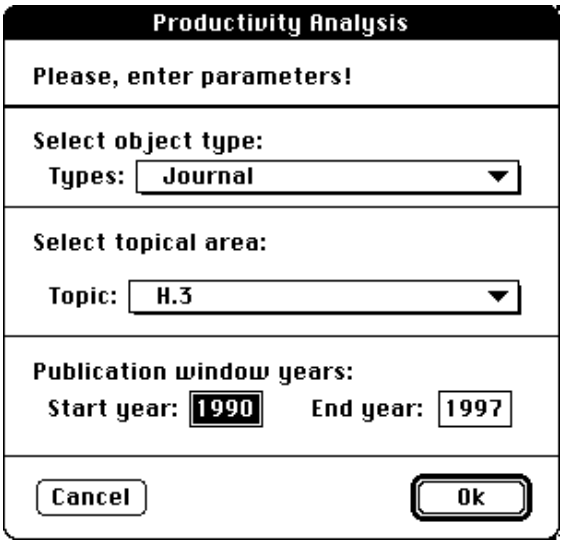


Fig. 4.5 (a) The online dialog for standard journal productivity

```
{(Information Processing and Manageme, 2),
(JASIS, Journal of the American Soci, 3)}
```

Fig. 4.5 (b) Sample result for journal productivity

The underlying query expands the selected domain to all of its subclasses and then finds the articles in this expanded domain published in the required time range. It then counts the number of articles for each journal (or other selected object type) in each of the classes. The query uses transitive relationships in a classification hierarchy. Instead of listing all possible subclasses of the class 'H.3' (for information retrieval), the query simply asks for all subclasses of 'H.3' by the expression *successors*('H.3', [SUBCLASS]) (see Järvelin & Niemi, 1997 for details).

The result data, Figure 4.5(b), are structured by the form JOURNAL(journal, art_cnt) and report all journals producing articles within the information retrieval area in the 1990's. Thus JASIS has produced 3 articles according to the database. This particular analysis, focusing on

journals, can also be done directly by means of the Dialog RANK command in the ISI databases and is one of the few cases where contemporary online systems are at the level of our approach.

Figure 4.6 gives the same result data organized by countries instead of journals. The online dialog is the same except for selecting the object type 'Country' for analysis instead of journals.

```
{(USA, 1),
 (Canada, 2),
 (Finland, 2)}
```

Fig. 4.6. Sample result for country productivity

4.3. Author co-citation analysis

Author co-citation analysis (ACA) is an established area of informetrics (e.g., [5]). McCain [35] gives a technical overview of the procedures required in ACA. In a traditional ACA data collection, co-citation counts are collected for each pre-selected pair of authors through a range of separate queries. These co-citation counts are then arranged into a raw co-citation matrix for further analysis, for instance, in order to generate maps of a scientific domain by means of multi-dimensional scaling (MDS). There are further complications in data collection if co-authors in the second author position or beyond are to be taken into account. In this section we demonstrate, how the raw data for ACA and its generalizations can be computed declaratively through the online dialogs of the FUN interface.

Figure 4.7(a) presents the online dialog for standard author co-citation analysis. In the dialog the user may choose author, institute or class co-citation analysis. The menu of ACM CS classes is produced automatically from the database and any class selections automatically include any subclasses into the analysis. The years of interest restrict the *publication years* of

the co-cited authors — the citations may come from later years. The result has the structure

AUTHOR_COCITATIONS(author, cc_author, cocicosum, sum_citing1, sum_citing2), where author and cc_author (after renaming) are the co-cited authors, cocicosum is the sum of co-citations to these authors, and sum_citing1 and sum_citing2 are the citation sums for the authors individually for all their articles published during the years of interest. The attribute 'cocicosum' gives the author co-citation strength as a simple sum of the authors' co-citations for their pairs of articles. The attributes sum_citing1 and sum_citing2 can be used to normalize the co-citation sum of the authors. The citation network CITES2 (excluding the self-citations) is used in the analysis. The query produces all co-cited author pairs within the selected domain (ACM CS Class H) without the user having to select the pairs individually.

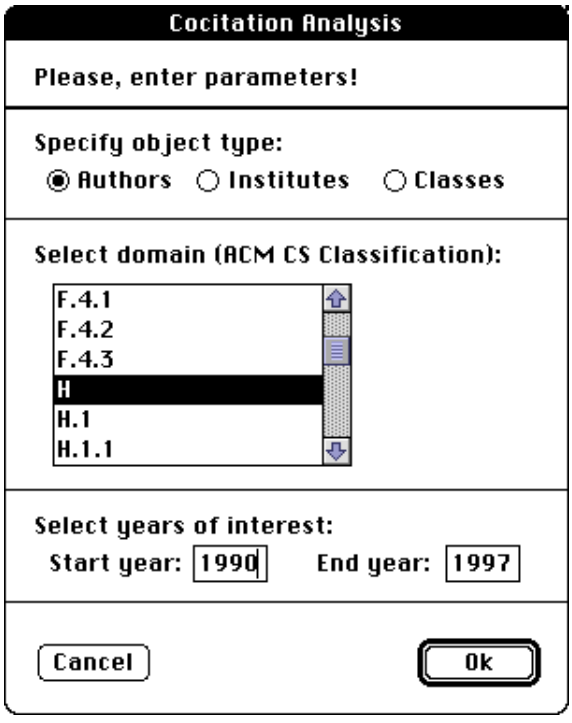


Fig. 4.7 (a) The online dialog for standard author cocitation analysis

Figure 4.7(b) presents a part of the resulting data which may be submitted to further ACA processing, e.g., for producing author clusters and maps. It is straightforward to use such data

as an input file to MDS for further analysis, as recently done on information science by White and McCain [36]. The sample data indicate that, for instance, Macleod and Lynch have been co-cited twice for their two and four individual citations, respectively.

Again, salient features of our expressions are, among others, that the user need not form retrieved sets for each cited author in advance and then produce the co-citation data for each pair of authors separately. Instead, the co-cited authors are found within the data. Moreover, *all authors* of cited papers are treated equally. Modeling article authors as an atomic-valued first author and a relation-valued 'COAUTHOR' set will make way for the traditional way of ACA, that is, focusing on first authors.

```
{…,
 (Lynch, C, Macleod, I,    2, 4,
2),
 (Lynch, C, Desai, B,      1, 2,
2),
 (Lynch, C, Sadri, F,      1, 2,
2),
 (Lynch, C, Goyal, P,      1, 2,
2),
 (Lynch, C, Scheck, H,     1, 2,
1),
 (Lynch, C, Scholl, M,     1, 2,
1),
 (Lynch, C, Kircz, J,      1, 2,
1),
 (Lynch, C, Bleeker, J,    1, 2,
1),
 (Macleod, I, Desai, B,    1, 1,
2),
 (Macleod, I, Sadri, F,    1, 1,
2),
 (Macleod, I, Goyal, P,    1, 1,
2),
 (Macleod, I, Scheck, H,   1, 1,
1),
 (Macleod, I, Scholl, M,   1, 1,
1),
 (Macleod, I, Kircz, J,    1, 1,
1),
 (Macleod, I, Bleeker, J,  1, 1,
1),
 (Macleod, I, Lynch, C,    2, 2,
4),
…}
```

Fig. 4.7 (b) Sample Expression 2 result for author co-citation analysis (partial)

4.3.1 Institutional and class co-citation

In the FUN interface, the user can easily navigate in the data structures and produce the data breakdown and aggregations relevant in her current situation. For example, she obtains institutional co-citation data simply by replacing the authors by their institutions in the **form** constructs. The result then has the structure *ORG_COCITATIONS(organization, cc_organization,* cocicosum, sum_citing1, sum_citing2) — modified attributes in italics. Selecting the radio button 'Institutes' in the online dialog, Figure 4.8(a), performs this replacement. The result of this query is given in Figure 4.8(b), which shows that Queens University and Concordia University have been co-cited three times for their publications in the CS domain 'H' in the 1980's. Class co-citations (e.g., for similarity analysis) are obtained through the radio button 'Classes'. Other frequently needed co-citation analyses may be produced by minor modifications of the underlying queries and by adding new radio buttons.

4.4. Keyword profiles of cited objects

White [5] mentions the possibility of replacing author points in an author co-citation map by three or four expressions appearing most frequently in the titles of articles *citing* each author. One may say that these expressions reflect, statistically, the issues and topics for which each author has produced a *recognized contribution*. In this section we demonstrate, how such information may be computed in the FUN interface.

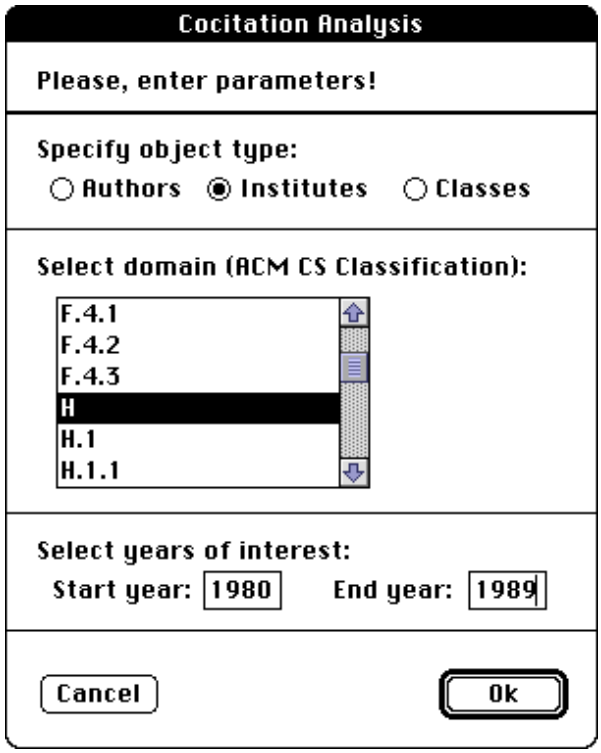


Fig. 4.8 (a) The online dialog for standard institutional co-citation analysis

```
{(Concordia University, Queens University,            3, 4,
8),
 (Concordia University, Elsevier Science Publishers, 1, 2,
1),
 (Elsevier Science Publishers, Concordia University, 1, 1,
2),
 (Elsevier Science Publishers, Queens University,     2, 2,
8),
 (Queens University, Concordia University,            3, 8,
4),
 (Queens University, Elsevier Science Publishers,     2, 8,
2)}
```

Fig. 4.8 (b) Sample result for institutional co-citation data (partial)

We illustrate White's idea by using *the keywords* of citing articles as content indicators for author contributions. In the online dialog (omitted here) the user needs only to select author names and the citation window as in the examples above. The query expression executing the analysis finds the articles by the selected authors and then, for each article, the citing articles and their keywords. These citing keywords are then counted for each author. The sample result is displayed in Figure 4.9.

The result (Figure 4.9) has the structure AUTHOR_CONTRIBUTION(author, CONTRIB_KEYS(citing_keyword, key_count)), giving for each cited author the citing keywords and their frequency 'key_count' summed from any articles citing any of his/her articles. The query result informs that, e.g., Crawford is known for contributions in relational databases and document retrieval.

As above, it is straightforward to obtain similar figures for journals, institutions or countries by simple modifications in the **form** construct. We have also defined online dialogs for standard recognized contribution analysis for these cases (bypassed here). In a very similar way one may compute the *scientific export* and the *geographical knowledge export* to other fields [4, 32]. The scientific export is calculated as follows:

```
{(Crawford, R,
     {(hierarchical objects, 1),
      (SGML, 1),
      (structured documents, 1),
      (query languages, 1),
      (bibliographic database, 1),
      (lazy evaluation, 1),
      (nonmaterialized relation, 1),
      (SQL, 1),
      (data restructuring, 1),
      (NF2 database, 1),
      (nf2 relation, 1),
      (query formulation, 1),
      (nest operation, 1),
      (non-first normal form relation, 1),
      (text retrieval, 2),
      (document retrieval, 2),
      (universal relation, 2),
      (relational database, 3)}),
 (Desai, B,
     {(data restructuring, 1),
      …}),
 (Macleod, I,
     {(data restructuring, 1),
      (document retrieval, 1),
      (NF2 database, 1),
      …})}
```

Fig. 4.9 Sample result – keyword profiles - for recognized contributions (partial)

- by selecting top journals in some area and finding all the source articles in these journals;
- by finding other articles which cite the source articles;

- by checking the classification codes or keywords of the citing articles (or of the journals, if available);
- by aggregating the classification codes or keywords.

The resulting sample list will look similar to Figure 4.9 in which the author names are replaced by journal names.

4.5. Geographical knowledge export

The geographical scatter of article users through citations is computed by the online dialog for standard geographical scatter of citations to journals, Figure 4.10(a). The analysis is similar to the analysis of recognized contribution (above). The user selects any set of journals from the journal name menu and the citation window. For each article published in each journal the underlying query finds the citing articles within the given citation window and their author home countries (citing countries). It then aggregates the number of citing articles per citing country.

The result is given in Figure 4.10(b). It has the structure JOURNAL(journal, CITINGCOUNTRIES(c_country, citation_sum)). For example, JASIS has received one citation from USA and four from Canada and Finland in the sample database.

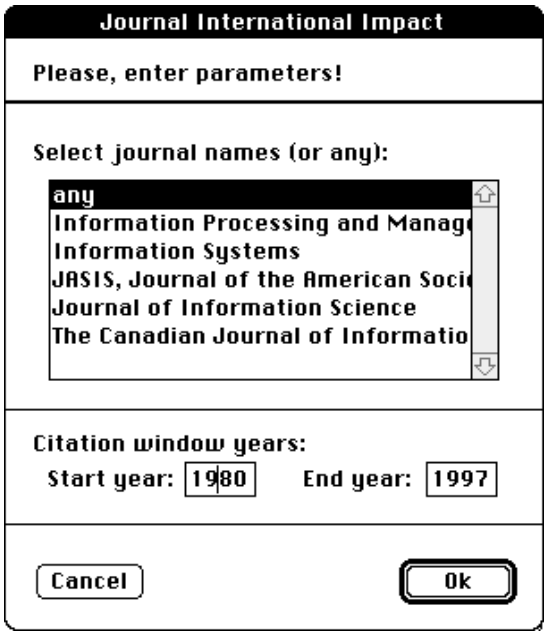


Fig. 4.10 (a) The online dialog for standard geographical scatter of citations to journals

```
{(Information Processing and Manageme...,
        {(Finland, 3)}),
 (Information Systems,
        {(Canada, 2),
         (Finland, 3)}),
 (JASIS, Journal of the American Soci...,
        {(USA, 1),
         (Canada, 4),
         (Finland, 4)}),
 (Journal of Information Science,
        {(Finland, 1)}),
 (The Canadian Journal of Information...,
        {(Finland, 1),
         (Canada, 1)})}
```

Fig. 4.10 (b) Sample result on geographical scatter of citations

It is straightforward to obtain similar result for other objects of interest, e.g., authors, institutions or countries, by very simple modifications in the **form** constructs which are easily implemented as online dialogs. Current online methods require treatment journal by journal or author by author. In the FUN interface, again, the statistics are obtained for all relevant journals (or other object types) at once.

5. DISCUSSION

The benefits of the proposed data modeling and query interface are methodological and conceptual. *Methodologically,* data for basic informetric concepts, such as impact factors, author co-citation analysis, international impact, productivity calculations in a given area, etc., can be computed easily and often with much less effort than in contemporary online and offline retrieval systems. More precisely, the methodological benefits can be summarized as follows:

- The user need not process each object of analysis separately as in current online methods. The objects of analysis can be specified implicitly and declaratively by the **conditions** construct of the FUN query language or through the online dialogs. Explicit identification of relevant objects for the statistics may require considerable experience in the area under consideration.
- The user need not specify each year of citation in citation analyses separately as traditionally done. Instead, she gets data for all relevant years automatically.
- Multiple statistics may be computed at once by a single query. For example, one may compute in a single query for each journal in a research domain the number of articles per journal, the average number of references per article in the journal, and the average number of citations per article in the journal. Current online systems do not support multi-level multi-attribute data aggregation.
- In co-citation analysis, the pairs for which statistics are computed, are formed automatically. In ACA in particular, the user need not create and process each author pair separately as currently in online ACA analysis [35].
- New statistically based qualitative data can be computed. For example, the recognized contribution analysis extends citation analysis by reporting qualitative information based on keyword profiles of the citing documents as projected by [5].

- The user can easily change focus, the type of objects of interest in the analysis, between articles, journals, authors, departments, organizations or countries. This is done by simple modifications in the **form** constructs or, at the online dialog level, by selecting the object type in a menu. Breakdowns of data are easily available, for instance, by years or classes, simply by introducing appropriate relation-valued and atomic-valued attributes in the **form** construct. It is equally easy to analyze, for example, journals by year as it is to analyze years by journals. Thus any object types may form the units of analysis or serve as data breakdown dimensions. In current online systems, such analyses, if at all possible, would require identifying new objects and repeating manually the multiple step process for each (pair) of them [7, 22].
- The FUN user query language is at a very high abstraction level and highly declarative. Therefore the user need not specify explicitly any data restructuring operations. Also the construction of relation-valued attributes based on sub-queries is at a very high level. Our idea is that the user describes, declaratively, only the relationships among the source and result data. In contemporary online retrieval systems often a very low-abstraction level step-by-step procedure is required, whereas in many advanced database systems the skill requirements on behalf of the user are too demanding [25]. The online dialogs relieve the users from the burden of using the query language at all. However, no usability tests with real users have been done.

The benefits of the FUN-interface for informetrics are based on data modeling and the interface's general expressive power. The modeling of bibliographic data as complex objects, which explicitly specify atomic-valued attributes and relation-valued attributes, supports analysis and aggregation of all structural components. The modeling of thesauri and classifications as binary relations supports transitive processing, for instance, automatic query ex

pansion to broad topical areas. The modeling of citations as binary relations supports easy processing both toward cited documents and toward citing documents.

The FUN interface provides a general expressive power allowing data restructuring, aggregation, retrieval, and transitive processing declaratively at a high abstraction level [13, 15]. There are no limitations on the organization of the result object types from the available source relation attributes and derived attributes. By placing source relation attributes and derived attributes in suitably arranged relation-valued attributes, complex result objects can be organized and subdivided flexibly. This supports generalized informetrics. The FUN interface as such is a general-purpose interface, which may be applied also in many areas outside informetrics.

Conceptually, the interface also supports several fruitful generalizations of typical informetric measurements. Such generalizations are obtainable by substituting traditional foci of analysis, for instance journals, by other object types, such as authors, organizations, countries or classes of a classification scheme. Through sample expressions we have shown how impact factors, co-citation frequencies, internationalization statistics as well as productivity may be generalized from their traditional object types of analysis to any of the object types of journals, articles, authors, departments, organizations, countries, classes, or years. Both diachronic as well as synchronic analyses can be performed easily. These may be accompanied with statistical breakdowns based on any of the remaining object types. We believe that such analyses are needed in generalized informetrics. Moreover, the proposed interface improves, as a spin-off effect, the possibilities of utilizing citation data in information retrieval, following the overlap investigations by McCain [37] and Pao [38] within the cognitive framework [39].

Although individual researchers tend to be satisfied by looking up their own works and calculate their stand-alone citation impact, more comprehensive analyses are required by research managers and policy makers. The former individual document analyses are obviously interesting to carry out and indeed easily performed by the proposed NF^2-based informetric tool. The latter analyses, however, are mandatory as benchmarks if one wishes to compare actual citation impact of individuals with his own or similar departments elsewhere or compare departments internationally. Further, comparisons of impact between countries on specific research topics or fields are increasingly carried out at EU and OECD levels [40].

Despite of the many benefits, there are several *limitations and issues* that deserve attention. Although the FUN interface has been implemented in Prolog and runs on several platforms (PCs, Macintoshes and Unix machines), it is still rather *a computational prototype* than a fully developed software product. A product would require further developments in efficiency for large amounts of data, user interfaces, and concurrency support. The prototype is a main-memory oriented data management system written in Prolog. Thus its run time performance depends directly on the allocated main memory and processor speed. In these areas, and with current technology trends, the near future is promising. However, a commercial software product might run much faster when implemented in another language, e.g., Java. For the reasons presented, the data modeling and the FUN interface presented in this paper point out directions towards how online informetrics may be developed and how this depends on data management techniques. As the interface stands, it requires downloading of data from ISI and other online or CD-ROM databases, and conversion to the NF^2 relation representation. This can be automated by writing a reader for each supplier-specific data format. The ISI records should be linked to records from other online databases to complete the citation data by full bibliographic data.

Although the FUN interface provides very high-level declarative *queries,* these *are not always simple* and may require considerable thought on behalf of the user. However, this problem was removed by storing predefined and parameterized queries for use through simple online dialogs. The planning and execution process of the queries in the FUN interface is described generally in [14, 25] and in detail in [41]. Among these, [14, 41] cover the full expressive power involving both NF^2 relational and transitive processing.

Data quality in source databases is a problem for all informetric analyses [7, 22]. They have pointed out several problems in online data set creation for informetric analysis:

- structural consistency of items within each database and between databases,
- availability of sufficient data in existing fields,
- consistency of coding of structural components (i.e., field tags),
- consistency of data item representation — e.g., how many different forms there are for person, journal or corporate names,
- consistency and quality of indexing and/or classification.

Lack of consistency and quality in these areas cause problems in data conversion from online databases to the NF^2 relation format of the FUN interface.

6. CONCLUSION

This article demonstrates how informetric calculations can be performed through modern data modeling techniques. The article is based on a small sample database and development of sample queries for informetric calculations. The queries are run through the FUN interface that is a computational prototype, which has been implemented in Prolog and runs on several platforms (PCs, Macintoshes and Unix machines). Therefore the data modeling and the FUN

interface presented in this paper point out directions how online informetrics may be developed and how this depends on data management techniques.

The article provides both methodological and conceptual contributions for informetrics. First, they are achieved through advanced data modeling of complex objects as well as terminological and citation networks, and secondly, through high-level declarative query interfaces providing a general expressive power allowing data restructuring, aggregation, retrieval, and transitive processing. In this way data for basic informetric concepts, such as bibliographic coupling, author co-citation analysis, impact factors, international visibility and international impact, productivity calculations in a given area, can be computed easily and often with much less effort than in contemporary online retrieval systems. Simultaneously, basic informetric concepts can also be generalized by substituting traditional foci of analysis, e.g., journals, by other object types, such as authors, organizations, countries or classes of a classification. There are no limitations on the organization of the result object types from the available source relation attributes and derived attributes. Statistical analyses for any of the object types may be refined by breakdowns based on any of the remaining object types. We believe that such analyses foster generalized informetrics.

REFERENCES

[1] Almind, T. and Ingwersen, P. Informetric analyses on the World Wide Web: Methodological approaches to "webometrics". *Journal of Documentation,* 53(4), 1997, 404-426.

[2] Persson, O. *BibExcel.* http://www.umu.se/inforsk/ (visited 10 March 1999).

[3] Egghe, L. and Rousseau, R. Introduction to Informetrics: Quantitative methods in Library, Documentation and Information Science. Amsterdam: Elsevier, 1990.

[4] Hjortgaard Christensen, F., Ingwersen, P. and Wormell, I. Online determination of the journal impact factor and its international properties. *Scientometrics,* 40(3), 1997, 529-540.

[5] White, H.D. ed., Perspectives on author co-citation analysis. *Journal of the American Society of Information Science,* 41(6), 1990, 430-468.

[6] Library Trends. Special issue on bibliometrics, summer 1981. *Library Trends,* 30(1).

[7] Hjortgaard Christensen, F. and Ingwersen, P. Online citation analysis: a methodological approach. *Scientometrics,* 37(1), 1996, 39-62.

[8] Sheck, H.-J. & Scholl, M.H. The relational model with relation-valued attributes. *Information Systems,* 11(2), 1986, 137-147.

[9] Deux, O. The story of O_2. IEEE Transactions on Knowledge and Data Engineering, 2(1), 1990, 91-108.

[10] Ullman, J.D. *Principles of database and knowledge base systems.* Vol. II. Rockville, MD: Computer Science Press, 1989.

[11] Paredaens, J., Peelman, P. and Tanca, L. G-log: A Graph-based query language. *IEEE Transactions on Knowledge and Data Engineering*, 7(3), 1995, 25-43.

[12] Desai, B.C., Goyal, P. and Sadri, F. Non-first normal form universal relations: An application to information retrieval systems. *Information Systems, 12*(1), 1987, 49 - 55.

[13] Agrawal, R., Borgida, A. and Jagadish, H.V. (1989). Efficient management of transitive relationships in large data and knowledge bases. In: Clifford, J. & al., eds. *The ACM Sigmod Conference,* Portland, Oregon, May 31-June 2, 1989. New York, NY: ACM Press, 1989, 253-262.

[14] Järvelin, K. and Niemi, T. Integration of complex objects and transitive relationships for information retrieval. *Information Processing & Management,* 35(5), 1999, xx-yy.

[15] Niemi, T. and Järvelin, K. Operation-Oriented Query Language Approach for Recursive Queries – Part 2. Prototype Implementation and Its Integration with Relational Databases. *Information Systems,* 17(1), 1992, 77-106.

[16] Järvelin, K. and Niemi, T. An NF^2 relational interface for document retrieval, restructuring and aggregation. In: Fox, E., Ingwersen, P. and Fidel, R., eds. *The 18th International Conference on Research and Development in Information Retrieval (ACM SIGIR '95),* Seattle, Wa, July 9-12, 1995. New York, NY: ACM Press, 1995, 102-110.

[17] Macleod, I. A. Storage and retrieval of structured documents. *Information Processing and Management,* 26(2), 1990, 197-208.

[18] Rada, R., Wang, W. and Birchall, A. Retrieval hierarchies in hypertext. *Information Processing and Management,* 29(3), 1993, 356-371.

[19] Salminen, A., Tague-Sutcliffe, J. and McClellan, C. From text to hypertext by indexing. *ACM Transactions on Information Systems, 13*(1), 1995, 69-99.

[20] Dialog. Get results with the Dialog RANK command. *Dialog Chronolog,* 21(1), 1993, 27-33.

[21] Ingwersen, P. A cognitive view of three selected online search facilities. *Online Review,* 8(5), 1984, 465-492.

[22] Ingwersen, P. and Hjortgaard Christensen, F. Data set isolation for bibliometric online analyses of research publications: Fundamental methodological issues. *Journal of the American Society for Information Science,* 48(3), 1997, 205-217.

[23] Järvelin, K. and Niemi, T. Deductive information retrieval based on classifications. *Journal of the American Society for Information Science,* 44(10), 1993, 557-578.

[24] Smith, J. and Smith, D. Database abstractions: Aggregation and generalization. *ACM Transactions on Database Systems,* 2(2), 1977, 105-133.

[25] Niemi, T. and Järvelin, K. A Straightforward NF^2 relational interface with applications in information retrieval. *Information Processing & Management,* 31(2), 1995, 215-231.

[26] Ullman, J.D. *Principles of database and knowledge base systems. Vol. I.* Rockville, MD: Computer Science Press, 1988.

[27] Pistor, P. and Andersen F. Designing a generalized NF^2 model with an sql-type language interface. In: Chu W. *& al.,* eds. *The 12th VLDB Conference,* Kyoto, Japan, August 21-23, 1986. Los Altos, CA: Morgan Kaufman, 1986, 278-285.

[28] Roth, M.A., Korth, H.F. and Batory, D.S. SQL/NF: a query language for ¬1NF relational databases. *Information Systems,* 12(1), 1987, 99-114.

[29] Südkamp, N. and Linnemann, V. Elimination of views and redundant variables in an SQL-like database language for extended NF^2 structures. In: D. McLeod *& al.*, eds. *Proceedings of the 16th VLDB Conference.* Palo Alto, CA: Morgan Kaufman Publishers, 1990, 302-313.

[30] Niemi, T. and Järvelin, K. The processing strategy for the NF^2 relational FRC-interface. *Information & Software Technology, 38,* 1996, 11-24.

[31] Moed, H.F. and van Leeuwen, Th.N. Impact factors can mislead. *Nature,* 381, 1996, 186.

[32] Wormell, I. Informetric analysis of the international impact of scientific journals: How international are the international journals? *Journal of Documentation,* 54(5), 1998, 584-605.

[33] Järvelin, K., Ingwersen, P. and Niemi, T. *Informetrics through advanced data management: Complex object restructuring, data aggregation and transitive computation.* Tampere, Finland: University of Tampere, Dept. of Information Studies, Report RN-1999-1, 1999. 42 p.

[34] Drott, P. Bradford's law: Theory, empiricism and gaps between. *Library Trends,* 30(1), 1981, 41-52.

[35] McCain, K.W. Mapping authors in intellectual space: a technical overview. *Journal of the American Society of Information Science,* 41(6), 1990, 433-443.

[36] White, H.D. and McCain, K.W. Visualizing a discipline: An author co-citation analysis of information Science, 1972-95. *Journal of American Society for Information Science,* 49(4), 1998, 327-355.

[37] McCain, K.W. Descriptor and citation retrieval in the medicine behavioural sciences literature: Retrieval overlaps and novelty. *Journal of American Society for Information Science,* 40, 1989, 110-114.

[38] Pao, M.L. Relevance odds of retrieval overlaps from seven search fields. *Information Processing & Management,* 30(3), 1994, 305-314.

[39] Ingwersen, P. Cognitive perspectives of information retrieval interaction: Elements of a cognitive IR theory. *Journal of Documentation*, 52(1), 1996, 3-50.

[40] May, Robert M. The scientific wealth of nations. *Science,* 275, 1997, 793-796.

[41] Järvelin, K. & Niemi, T. *Integration of complex objects and transitive relationships for information retrieval.* Tampere, Finland: University of Tampere, Department of Computer Science, Report A-1997-11. 61 p.